

## Method & Utilities

The e-government mobile standard framework provides methods and properties for \$.mobile object.

### \$.mobile.changePage (method)

This is a method to change pages. It is internally used to click a link or transfer a form in an environment where screen change and page loading are supported.

---

## Arguments

### to (string or object, required)

String: absolute or relative URL("about/us.html")

Object: jQuery collection object. (\$("#about"))

### options (object, optional)

Properties:

- transition (string, default: \$.mobile.defaultPageTransition): screen transition to show pages.
- reverse (boolean, default: false): decides the direction of screen transition.
- changeHash (boolean, default: true): decides whether to update the hash in the address box.
- role (string, default: undefined): data-role value used to print out pages on the screen. It is dependant on @data-role properties. The default value is 'undefined'.
- pageContainer (jQuery collection, default: \$.mobile.pageContainer): identifies elements that should contain pages.
- type (string, default: "get"): identifies whether the method is 'get' or 'post' type when making a page request. (used when the to argement of changePage() is URL.)
- data (object or string, default: undefined): data to be delivered with Ajax page request. (used when the to argement of changePage() is URL.)
- reloadPage (boolean, default: false): reload pages. Including the pages in DOM of the page container. (used when the to argement of changePage() is URL.)

---

## Examples

```
//transition to the "about us" page with a slideup transition
$.mobile.changePage( "about/us.html", { transition: "slideup" } );

//transition to the "search results" page, using data from a form with an ID of "search"
$.mobile.changePage( "searchresults.php", {
  type: "post",
  data: $("#form#search").serialize()
});

//transition to the "confirm" page with a "pop" transition without tracking it in history
$.mobile.changePage( "../alerts/confirm.html", {
  transition: "pop",
  reverse: false,
```

```
changeHash: false
});
```

## **\$.mobile.loadPage (method)**

Load outside pages and add them to DOM. This method is internally called by `changePage()` function when URL comes as the first argument. This function does not affect the currently activated pages. It is used when loading pages in the background. It returns pre-defined objects after pages are loaded and documents are added.

### **Arguments**

**url (string, required) relative or absolute URL**

**options (object, optional)**

Properties:

- `role` (string, default: undefined) ): data-role value used to print out pages on the screen. It is dependant on `@data-role` properties. The default value is 'undefined'.
- `pageContainer` (jQuery collection, default: `$.mobile.pageContainer`): identifies elements that should contain pages..
- `type` (string, default: "get"): identifies whether the method is 'get' or 'post' type when making a page request.
- `data` (object or string, default: undefined): data to be delivered with Ajax page request.
- `reloadPage` (boolean, default: false): reload pages. Including the pages in DOM of the page container.

### **Examples**

```
//load the "about us" page into the DOM
$.mobile.loadPage( "about/us.html" );
```

```
//load a "search results" page, using data from a form with an ID of "search"
$.mobile.loadPage( "searchresults.php", {
  type: "post",
  data: $("form#search").serialize()
});
```

## **jqmData(), jqmRemoveData(), and jqmHasData() (method)**

### **Arguments**

`jqmData()` and `jqmRemoveData()` are the data of jquery core and `removeData`. Setting and getting of data property are automatically created.

---

## Examples

---

`$("#div:jqmData(role='page')")` or `$("#div[data-"+ $.mobile.ns +"role='page']")` may be used in stead of `$("#div[data-role='page']")`.

### **\$.mobile.showPageLoadingMsg ()**

---

Shows the page loading message.

---

## Examples

---

```
//cue the page loader
$.mobile.showPageLoadingMsg();
```

### **\$.mobile.hidePageLoadingMsg ()**

---

Hides the page loading message.

---

## Examples

---

```
//cue the page loader
$.mobile.hidePageLoadingMsg();
```

### **\$.mobile.path.parseUrl (method)**

---

This is a function that parses URLs. It make access to URL's elements(data) easier. If there is no value during parsing, use empty string to fill items (e.g. protocol, host, etc.). When parsing URLs which don't hava authority, data is included after protocol/scheme colon.

---

## Arguments

---

**url (string, required) relative or absolute URL**

### **Return Value**

Returns objects that include various URL components of text string format. The property of the object follows the location object.

### **hash**

`#` fragment of URL component that includes characters.

### **host**

Host and port number of URL.

### **hostname**

Host name included in URL.

**href**

Original copy of analyzed URL.

**pathname**

Path of directory or file referenced by URL.

**port**

A port identified within URL. The default port can be used by the used protocol and most are displayed in empty strings.

**protocol**

Protocol that supports URL that includes ':' character.

**search**

Query component of URL that includes '?' character. The `$.mobile.path.parseUrl()` method does not only provide URLs of key formats accessible by the developer, but also properties that support access to other components.

**authority**

Username, password and host component of URL.

**directory**

Directory component of pathname (without the file name)

**domain**

Component related with protocol and authority of URL.

**filename**

File name within the pathname (except directory).

**hrefNoHash**

Original copy of URL except the fragment(hash) component.

**hrefNoSearch**

Original copy of URL except query(search) and fragment(hash).

**password**

Password included the authority component.

## username

Username included the authority component.

---

## Examples

---

```
// Parsing the Url below results an object that is returned with the
// following properties:
//
// obj.href:
http://jblas:password@mycompany.com:8080/mail/inbox?msg=1234&type=unread#msg-content
// obj.hrefNoHash:
http://jblas:password@mycompany.com:8080/mail/inbox?msg=1234&type=unread
// obj.hrefNoSearch: http://jblas:password@mycompany.com:8080/mail/inbox
// obj.domain: http://jblas:password@mycompany.com:8080
// obj.protocol: http:
// obj.authority: jblas:password@mycompany.com:8080
// obj.username: jblas
// obj.password: password
// obj.host: mycompany.com:8080
// obj.hostname: mycompany.com
// obj.port: 8080
// obj.pathname: /mail/inbox
// obj.directory: /mail/
// obj.filename: inbox
// obj.search: ?msg=1234&type=unread
// obj.hash: #msg-content

var obj =
$.mobile.path.parseUri("http://jblas:password@mycompany.com:8080/mail/inbox?msg=
1234");
```

---

## **\$.mobile.path.makePathAbsolute (method)**

---

Method utility function that changes a relative path to an absolute path.

---

## Arguments

---

relPath (string, required); relative path of a file or directory

absPath (string, required) An absolute file or relative path to resolve against.

---

## Return Value

---

Returns the absolute path as a text string.

---

## Examples

---

```
// Returns: /a/b/c/file.html
var absPath = $.mobile.path.makePathAbsolute("file.html", "/a/b/c/bar.html");

// Returns: /a/foo/file.html
var absPath = $.mobile.path.makePathAbsolute(".././foo/file.html", "/a/b/c/bar.html");
```

### **\$.mobile.path.makeUrlAbsolute (method)**

Utility method that changes a relative path URL to an absolute path URL.

---

### Arguments

---

relUrl (string, required): relative URL  
absUrl (string, required) An absolute URL to resolve against.

---

### Return Value

---

Returns the absolute path as a text string.

---

## Examples

---

```
// Returns: http://foo.com/a/b/c/file.html
var absUrl = $.mobile.path.makeUrlAbsolute("file.html", "http://foo.com/a/b/c/test.html");

// Returns: http://foo.com/a/foo/file.html
var absUrl = $.mobile.path.makeUrlAbsolute(".././foo/file.html", "http://foo.com/a/b/c/test.html");

// Returns: http://foo.com/bar/file.html
var absUrl = $.mobile.path.makeUrlAbsolute("//foo.com/bar/file.html",
"http://foo.com/a/b/c/test.html");

// Returns: http://foo.com/a/b/c/test.html?a=1&b=2
var absUrl = $.mobile.path.makeUrlAbsolute("?a=1&b=2", "http://foo.com/a/b/c/test.html");

// Returns: http://foo.com/a/b/c/test.html#bar
var absUrl = $.mobile.path.makeUrlAbsolute("#bar", "http://foo.com/a/b/c/test.html");
```

### **\$.mobile.path.isSameDomain (method)**

Compares two URL and checks if they are the same domains, and then return Boolean values.

---

## Arguments

---

url1 (string, required) relative URL  
url2 (string, required) absolute URL (An absolute URL to resolve against.)

---

## Return Value

---

If the domains are the same, true is returned. If not, false is returned.

---

## Examples

---

```
// Returns: true
var same = $.mobile.path.isSameDomain("http://foo.com/a/file.html",
"http://foo.com/a/b/c/test.html");

// Returns: false
var same = $.mobile.path.isSameDomain("file://foo.com/a/file.html",
"http://foo.com/a/b/c/test.html");

// Returns: false
var same = $.mobile.path.isSameDomain("https://foo.com/a/file.html",
"http://foo.com/a/b/c/test.html");

// Returns: false
var same = $.mobile.path.isSameDomain("http://foo.com/a/file.html",
"http://bar.com/a/b/c/test.html");
```

---

## **\$.mobile.path.isRelativeUrl (method)**

---

Checks if URL is a relative path and returns Boolean value.

---

## Arguments

---

url (string, required) relative or absolute URL

---

## Return Value

---

If it is a relative URL, true is returned. If it is an absolute URL, false is returned.

---

## Examples

---

```
// Returns: false
var isRel = $.mobile.path.isRelativeUrl("http://foo.com/a/file.html");
```

```
// Returns: true
var isRel = $.mobile.path.isRelativeUrl("//foo.com/a/file.html");

// Returns: true
var isRel = $.mobile.path.isRelativeUrl("/a/file.html");

// Returns: true
var isRel = $.mobile.path.isRelativeUrl("file.html");

// Returns: true
var isRel = $.mobile.path.isRelativeUrl("?a=1&b=2");

// Returns: true
var isRel = $.mobile.path.isRelativeUrl("#foo");
```

### **\$.mobile.path.isAbsoluteUrl (method)**

Checks if URL is an absolute path and returns Boolean value.

---

#### **Arguments**

url (string, required) relative or absolute URL

---

#### **Return Value**

If it is an absolute URL, true is returned. If it is a relative URL, false is returned.

---

#### **Examples**

```
// Returns: true
var isAbs = $.mobile.path.isAbsoluteUrl("http://foo.com/a/file.html");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("//foo.com/a/file.html");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("/a/file.html");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("file.html");

// Returns: false
```

```
var isAbs = $.mobile.path.isAbsoluteUrl(s"?a=1&b=2");

// Returns: false
var isAbs = $.mobile.path.isAbsoluteUrl("#foo");
```

## **\$.mobile.base (methods, properties)**

Ensures that it works as a generalized base element.

## **\$.mobile.silentScroll (method)**

Scrolls to Y coordinate designated as an argument without running the scroll event listener.

### **Arguments**

yPos (number, defaults to 0). Y coordinates for scroll

### **Examples**

```
//scroll to Y 100px
$.mobile.silentScroll(100);
```

## **\$.mobile.addResolutionBreakpoints (method)**

Set the range for the minimum and maximum widths of the class to be added to HTML elements. Adds breakpoints to the added min/max width classes.

### **Arguments**

Conveys numbers or arrays to be added to values (number or array value) and resolution class. For details, refer to Orientation & resolution targeting.

### **Examples**

```
//add a 400px breakpoint
$.mobile.addResolutionBreakpoints(400);
//add 2 more breakpoints
$.mobile.addResolutionBreakpoints([600,800]);
```

## **\$.mobile.activePage (property)**

Returns the reference of the current page.